



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/749,615	12/30/2003	Gregor K. Frey	6570P031	8289
45962	7590	12/23/2008		
SAP/BSTZ BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP 1279 OAKMEAD PARKWAY SUNNYVALE, CA 94085-4040			EXAMINER LINDSEY, MATTHEW S	
			ART UNIT	PAPER NUMBER
			2451	
			MAIL DATE	DELIVERY MODE
			12/23/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/749,615

Applicant(s)

FREY ET AL.

Examiner

MATTHEW S. LINDSEY

Art Unit

2451

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 16 October 2008.
2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-30 is/are pending in the application.
4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5) ☐ Claim(s) _____ is/are allowed.
6) ☒ Claim(s) 1-30 is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SI/309)
Paper No(s)/Mail Date 10/16/2008
4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____

DETAILED ACTION

1. Claims 1-30 are pending in this application. Claims 1, 10, 19 and 28 have been amended as filed 16 October 2008.

Continued Examination Under 37 CFR 1.114

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 16 October 2008 has been entered.

Claim Objections

3. Claim 10 is objected to because of the following informalities: Claim 10 recites: "instances a tracing controllers" (Claim 10, line 22) and "instances a logging controllers" (Claim 10, line 23). For the purposes of examination these will be treated as: "instances of tracing controllers" and "instances of logging controllers".
4. Claim 19 is objected to because of the following informalities:

Claim 19 includes a double colon; see Claim 19, line 3, where it is apparent that a single colon was intended.

Furthermore, Claim 19 recites: "said first and second controllers [[each being: [[an]] instances]" (Claim 19, lines 14-15). For the purpose of examination this will be treated as "said first and second controllers [[each]] being: [[an]] instances".

Appropriate correction is required.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 19-30 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

7. With respect to Claims 19-27, the claims are directed to "A system" but fail to recite any physical components and as such can be reasonably interpreted as software. For example see the instant specification [0019] – [0022], specifically: "One embodiment of the invention is implemented in an object oriented programming environment such as Java" ([0019], lines 7-8), and "each controller 200 is an instance of a defined 'controller' class" ([0020], lines 1-2). The claims lack the physical objects or articles to constitute a machine or manufacture under the meaning of 35 USC 101.

They are not a series of acts or steps to be a process, nor are they a combination of chemical compounds to be a composition of matter. They represent software and as such fail to fall within a statutory category of invention.

8. With respect to Claims 28-30 they are directed to "An article of manufacture having program code stored thereon". The instant specification does not clearly define article of manufacture, but does define computer readable media to include signals and/or carrier waves, see for example, the instant specification at [0090], where: "the present invention may be downloaded as a computer program which may be transferred from a remote computer... to a requesting computer... by way of data signals embodied in a carrier wave or other propagation medium" ([0090], lines 6-9). As such the claim is drawn to a form of energy. Energy is not one of the four categories of invention and therefore these claims are not statutory. Energy is not a series of steps or acts to be a process. Energy is not a physical article or object and as such is not a machine or manufacture. Energy is not a combination of substances and therefore not a composition of matter.

Claim Rejections - 35 USC § 112

9. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

10. Claims 1, and 10 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

11. Claims 1 and 10 recite the limitation "said determining" in Claim 1, line 11; Claim 10, line 11. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 103

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

13. Claims 1-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shah (US 2003/0005173), hereinafter Shah, in view of Ullman et al. (US 7,120,685 B2), hereinafter Ullman.

14. With respect to Claim 1, Shah disclosed: "A method (Abstract, lines 1-2) comprising:

defining a severity level for a first controller ([0050], lines 1-8, and [0054], lines 5-7, where the severity level is the configuration of the filter, for example the amount of data passed to handlers from the filter can change based on the filter configuration and

thus the severity level is defined), the first controller processing messages based on the effective severity level ([0050], lines 2-4)",

"the second controller being a child controller to the first controller in a controller hierarchy ([0059], lines 3-5) so as to receive said security level as a parameter from said first controller ([0059], lines 5-9, if a property on a child is not set, it inherits that property from the parent, and for example, trace node has the property isLogging=false, both of its children inherit this property); and

setting the severity level of the second controller equal to the severity level of the first controller ([0059], lines 3-7) if the severity level of the first controller is between the minimum severity level ([0059], lines 7-8, isLogging=false, where no logging takes place) and the maximum severity level ([0059], second Col., lines 2-4, isLogging=true, where logging takes place), said first and second controllers each being:

tracing controller instances ([0066], lines 8-10, where the child trace node can be configured independently of its parent trace node); or,

logging controller instances ([0068], lines 1-3, where message loggers can be similarly configured)".

Shah did not explicitly state: "wherein, available severity levels include: a) a first severity level that indicates the existence of an anomaly that an application can recover from, said application also able to perform a desired task; b) a second severity level that indicates the existence of an error that an application can recover from, said application also being unable to perform a desired task; wherein, said determining includes

selecting one of a), b) above; defining a minimum severity level and a maximum severity level with a second controller".

However, Ullman disclosed: "wherein, available severity levels include: a) a first severity level that indicates the existence of an anomaly that an application can recover from, said application also able to perform a desired task (Col. 3, lines 53-60, specifically warning messages, e.g., 'The available RAM is dangerously low' indicates the existence of an anomaly that an application can recover from and still perform a desired task);

b) a second severity level that indicates the existence of an error that an application can recover from (Col. 1, lines 29-33, where an example error is a misspelling on the input, and applications can recover from user misspelling on input), said application also being unable to perform a desired task (Col. 3, lines 53-60, specifically error messages, e.g., 'An error has occurred in this program' indicates the desired task had an error);

wherein, said determining includes selecting one of a), b) above (Col. 3, lines 53-60, where high level messages include the warning messages and error messages);

defining a minimum severity level and a maximum severity level with a second controller (Abstract, lines 9-15, and Col. 4, lines 15-29, where a maximum level is defined when selective control of message logging, tracing, and filtering of the logger and tracer output are at the maximum level, for example when the frequency of logging is adjusted upward the logging level will increase to its maximum level. During normal

operation or after detection of a stop event the frequency of logging may be decreased, for a minimum level)".

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the logging system of Shah with the teachings of Ullman to include support for variable levels of logging. Motivation to combine these references comes from Ullman, "to provide continual monitoring with logging and tracing of all computing system activities is unduly burdensome and adversely impacts performance objectives. Therefore, what is provided is a variable logging frequency monitor feature which implements selective control of the message logging, the tracing, and of the filtering of the logger and tracer output (Col. 4, lines 12-18)". Therefore by combining the logging system of Shah with the variable levels of logging of Ullman, one can save performance while still logging relevant events on a computing system.

15. With respect to Claim 10, Shah disclosed: "A method (Abstract, lines 1-2) comprising:

defining an severity level for a first controller ([0050], lines 1-4, and [0054], lines 5-7, where the severity level is the configuration of the filter, for example the amount of data passed to handlers from the filter can change based on the filter configuration and thus the severity level is defined), the first controller processing messages based on the defined severity level ([0050], lines 2-4)",

"the second controller being a child controller to the first controller in a defined controller hierarchy ([0059], lines 3-5); and

setting the severity of the second controller equal to the minimum severity level if the severity level of the first controller is relatively lower than the defined minimum severity level ([0059], lines 7-8, and second Col., lines 1-4, where the parent node, object 601 in Figure 6, has no logging, which is lower than the minimum defined logging level for the child node, object 602, Figure 6, which has logging set to true. The parent level is relatively lower than the child minimum level, and the child overrides the parent value) and setting the severity of the second controller equal to the maximum severity level if the severity level of the first controller is relatively higher than the defined maximum severity level (As seen in the rejection of the previous limitation, the child does not have to inherit the value of the parent, so it is conceivable that the parent has logging set to true and the child has logging set to false, which would mean the parent has a higher logging lever than the child, and the child will use its lower level rather than inherit the parent value), said first and second controllers being:

instances of tracing controllers ([0066], lines 8-10, where the child trace node can be configured independently of its parent trace node); or,

instances of logging controllers ([0068], lines 1-3, where message loggers can be similarly configured)".

Shah did not explicitly state: "wherein, available severity levels include: a) a first severity level that indicates the existence of an anomaly that an application can recover from, said application also able to perform a desired task; b) a second severity level that indicates the existence of an error that an application can recover from, said application also being unable to perform a desired task; wherein, said determining includes

selecting one of a), b) above; defining a minimum severity level and a maximum severity level with a second controller”.

However, Ullman disclosed: “wherein, available severity levels include: a) a first severity level that indicates the existence of an anomaly that an application can recover from, said application also able to perform a desired task (Col. 3, lines 53-60, specifically warning messages, e.g., ‘The available RAM is dangerously low’ indicates the existence of an anomaly that an application can recover from and still perform a desired task);

b) a second severity level that indicates the existence of an error that an application can recover from (Col. 1, lines 29-33, where an example error is a misspelling on the input, and applications can recover from user misspelling on input), said application also being unable to perform a desired task (Col. 3, lines 53-60, specifically error messages, e.g., ‘An error has occurred in this program’ indicates the desired task had an error);

wherein, said determining includes selecting one of a), b) above (Col. 3, lines 53-60, where high level messages include the warning messages and error messages);

defining a minimum severity level and a maximum severity level with a second controller (Abstract, lines 9-15, and Col. 4, lines 15-29, where a maximum level is defined when selective control of message logging, tracing, and filtering of the logger and tracer output are at the maximum level, for example when the frequency of logging is adjusted upward the logging level will increase to its maximum level. During normal

operation or after detection of a stop event the frequency of logging may be decreased, for a minimum level)".

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the logging system of Shah with the teachings of Ullman to include support for variable levels of logging. Motivation to combine these references comes from Ullman, "to provide continual monitoring with logging and tracing of all computing system activities is unduly burdensome and adversely impacts performance objectives. Therefore, what is provided is a variable logging frequency monitor feature which implements selective control of the message logging, the tracing, and of the filtering of the logger and tracer output (Col. 4, lines 12-18)". Therefore by combining the logging system of Shah with the variable levels of logging of Ullman, one can save performance while still logging relevant events on a computing system.

16. With respect to Claim 2 the combination of Shah and Ullman disclosed: "The method as in claim 1 further comprising: setting the severity level of the second controller equal to the minimum severity level if the severity level of the first controller is relatively lower than the defined minimum severity level (Shah, [0059], lines 7-8, and second Col., lines 1-4, where the parent node, object 601 in Figure 6, has no logging, which is lower than the minimum defined logging level for the child node, object 602, Figure 6, which has logging set to true. The parent level is relatively lower than the child minimum level, and the child overrides the parent value)".

17. With respect to Claim 11, the combination of Shah and Ullman disclosed: "The method as in claim 10 further comprising: setting the severity of the second controller equal to the severity of the first controller (Shah, [0059], lines 3-7) if the severity level of the first controller is between the minimum severity level (Shah, [0059], lines 7-8, isLogging=false, where no logging takes place) and the maximum severity level (Shah, [0059], second Col., lines 2-4, isLogging=true, where logging takes place)".

18. With respect to Claims 3 and 12, the combination of Shah and Ullman disclosed: "setting the severity level of the second controller equal to the maximum severity level if the severity level of the first controller is relatively higher than the defined maximum severity level (As seen in the rejection of Claim 2 above, the child does not have to inherit the value of the parent, so it is conceivable that the parent has logging set to true and the child has logging set to false, which would mean the parent has a higher logging lever than the child, and the child will use its lower level rather than inherit the parent value)".

19. With respect to Claims 4 and 13, the combination of Shah and Ullman disclosed: "wherein said first and second controllers are tracing controllers (Shah, [0066], lines 8-10, where the child trace node can be configured independently of its parent trace node), further comprising: receiving a trace message directed at the second controller (Shah, [0042], lines 2-4, and Figure 6, object 601, being the trace logger); comparing a severity value associated with the trace message with the severity level of the second

controller (Shah, [0050], lines 1-2 and 6-8); and processing the trace message if the severity value associated with the trace message is greater than or equal to the severity level of the second controller (Shah, [0050], lines 2-4).

20. With respect to Claims 5 and 14, the combination of Shah and Ullman disclosed: "wherein processing comprises formatting information contained in the trace message (Shah, [0051], lines 1-2) and forwarding the formatted information to a specified output destination (Shah, [0043], lines 1-5)".

21. With respect to Claims 6 and 15, the combination of Shah and Ullman disclosed: "wherein the trace message is emitted in response to the execution of a specified area of program code within an application (Shah, [0042], lines 6 and second Col., lines 1-6)".

22. With respect to Claims 7 and 16, the combination of Shah and Ullman disclosed: "wherein said first and second controllers are logging controllers (Shah, [0066], lines 8-10, where the child trace node can be configured independently of its parent trace node, and [0068], lines 1-3, where message loggers can be similarly configured) further comprising: receiving a log message directed at the second controller (Shah, [0042], lines 2-3); comparing a severity value associated with the log message with the severity level of the second controller (Shah, [0050], lines 1-2, and 6-8); and processing the log

message if the severity value associated with the trace message is greater than or equal to the severity level of the second controller (Shah, [0050], lines 2-4)".

23. With respect to Claim 8, the combination of Shah and Ullman disclosed: "wherein said first and second controllers are logging controllers (Shah, [0066], lines 8-10, where the child trace node can be configured independently of its parent trace node, and [0068], lines 1-3, where message loggers can be similarly configured) processing comprises formatting information contained in the log message (Shah, [0051], lines 1-2) and forwarding the formatted information to a specified output destination (Shah, [0043], lines 1-5)".

24. With respect to Claim 17, the combination of Shah and Ullman disclosed: "wherein processing comprises formatting information contained in a log message (Shah, [0051], lines 1-2) and forwarding the formatted information to a specified output destination (Shah, [0043], lines 1-5)".

25. With respect to Claims 9 and 18, the combination of Shah and Ullman disclosed: "wherein the second controller is associated with a particular category (Shah, [0042], lines 4-5) and wherein the log message is from an application and/or network component associated with the category (Shah, [0042], lines 4-5)".

26. Claims 19-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shah in view of Ullman and further in view of Biegel et al (US 5,608,720).

27. With respect to Claim 19, Shah disclosed: "A system (Abstract, lines 1-2) comprising:

a first controller ([0050], lines 1-4) to process messages based on a defined severity level ([0050], lines 2-4, and [0054], lines 5-7, where the severity level is the configuration of the filter, for example the amount of data passed to handlers from the filter can change based on the filter configuration and thus the severity level is defined), wherein, available severity levels include::

a) a first severity level that provides information for debugging ([0042], lines 6-14)"; and

"a second controller being a child controller to the first controller in a defined controller hierarchy ([0059], lines 3-5); and

severity generation logic to set the severity of the second controller equal to the severity of the first controller ([0059], lines 3-7) if the severity level of the first controller is between a minimum severity level ([0059], lines 7-8, isLogging=false, where no logging takes place) and a maximum severity level ([0059], second Col., lines 2-4, isLogging=true, where logging takes place), said first and second controllers being:

instances of tracing controllers ([0066], lines 8-10, where the child trace node can be configured independently of its parent trace node)".

Shah did not explicitly state: "b) a second severity level that indicates when a method is entered and/or exited; c) a third severity level that echoes what has been performed".

However, Ullman disclosed: "c) a third severity level that echoes what has already been performed (Col. 3, lines 60-61)".

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the logging system of Shah with the teachings of Ullman to include support for variable levels of logging. Motivation to combine these references comes from Ullman, "to provide continual monitoring with logging and tracing of all computing system activities is unduly burdensome and adversely impacts performance objectives. Therefore, what is provided is a variable logging frequency monitor feature which implements selective control of the message logging, the tracing, and of the filtering of the logger and tracer output (Col. 4, lines 12-18)". Therefore by combining the logging system of Shah with the variable levels of logging of Ullman, one can save performance while still logging relevant events on a computing system.

The combination of Shah and Ullman did not explicitly state: "b) a second severity level that indicates when a method is entered and/or exited".

However, Biegel disclosed: "b) a second severity level that indicates when a method is entered and/or exited (Col. 28, lines 45-49 and Col. 29, lines 16-20, where trace levels include Level 5, which include entry to functions or methods)".

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the logging system of Shah and Ullman with the teachings of Biegel to include support for tracing entry/exit from methods. Motivation to combine these comes from Biegel, where: "The trace Log contains software and system error conditions that are detected on a circuit packet. It aids in system debugging during integration and in the field by providing visibility into the execution flow through software entities" (Col. 29, lines 38-41). Therefore by combining the references one is aided in debugging by having visibility of the execution flow through software.

28. With respect to Claim 28, Shah disclosed: "An article of manufacture having program code stored thereon (Abstract, lines 1-2) which, when executed by a machine cause the machine to perform the operations of:

defining a severity level for a first controller ([0050], lines 1-4, and [0054], lines 5-7, where the severity level is the configuration of the filter, for example the amount of data passed to handlers from the filter can change based on the filter configuration and thus the severity level is defined), the first controller processing messages based on the defined effective severity level ([0050], lines 2-4), wherein, available severity levels include:

a) a first severity level that provides information for debugging ([0042], lines 6-14)"; and

"the second controller being a child controller to the first controller in a controller hierarchy ([0059], lines 3-5) so as to receive said security level as a parameter from

said first controller ([0059], lines 5-9, if a property on a child is not set, it inherits that property from the parent, and for example, trace node has the property isLogging=false, both of its children inherit this property); and

setting the severity level of the second controller equal to the severity level of the first controller ([0059], lines 3-7) if the severity level of the first controller is between the minimum severity level ([0059], lines 7-8, isLogging=false, where no logging takes place) and the maximum severity level ([0059], second Col., lines 2-4, isLogging=true, where logging takes place), said first and second controllers being:

instances of tracing controllers ([0066], lines 8-10, where the child trace node can be configured independently of its parent trace node)".

Shah did not explicitly state: "b) a second severity level that indicates when a method is entered and/or exited; c) a third severity level that echoes what has been performed; defining a minimum severity level and a maximum severity level with a second controller".

However, Ullman disclosed: "c) a third severity level that echoes what has already been performed (Col. 3, lines 60-61); defining a minimum severity level and a maximum severity level with a second controller (Abstract, lines 9-15, and Col. 4, lines 15-29, where a maximum level is defined when selective control of message logging, tracing, and filtering of the logger and tracer output are at the maximum level, for example when the frequency of logging is adjusted upward the logging level will

increase to its maximum level. During normal operation or after detection of a stop event the frequency of logging may be decreased, for a minimum level)".

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the logging system of Shah with the teachings of Ullman to include support for variable levels of logging. Motivation to combine these references comes from Ullman, "to provide continual monitoring with logging and tracing of all computing system activities is unduly burdensome and adversely impacts performance objectives. Therefore, what is provided is a variable logging frequency monitor feature which implements selective control of the message logging, the tracing, and of the filtering of the logger and tracer output (Col. 4, lines 12-18)". Therefore by combining the logging system of Shah with the variable levels of logging of Ullman, one can save performance while still logging relevant events on a computing system.

The combination of Shah and Ullman did not explicitly state: "b) a second severity level that indicates when a method is entered and/or exited".

However, Biegel disclosed: "b) a second severity level that indicates when a method is entered and/or exited (Col. 28, lines 45-49 and Col. 29, lines 16-20, where trace levels include Level 5, which include entry to functions or methods)".

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the logging system of Shah and Ullman with the teachings of Biegel to include support for tracing entry/exit from methods. Motivation to combine these comes from Biegel, where: "The trace Log contains software and system error

conditions that are detected on a circuit packet. It aids in system debugging during integration and in the field by providing visibility into the execution flow through software entities" (Col. 29, lines 38-41). Therefore by combining the references one is aided in debugging by having visibility of the execution flow through software.

29. With respect to Claims 20 and 29, the combination of Shah, Ullman and Biegel disclosed: "setting the severity level of the second controller equal to the minimum severity level if the severity level of the first controller is relatively lower than the defined minimum severity level (Shah, [0059], lines 7-8, and second Col., lines 1-4, where the parent node, object 601 in Figure 6, has no logging, which is lower than the minimum defined logging level for the child node, object 602, Figure 6, which has logging set to true. The parent level is relatively lower than the child minimum level, and the child overrides the parent value)".

30. With respect to Claims 21 and 30, the combination of Shah, Ullman and Biegel disclosed: "setting the severity level of the second controller equal to the maximum severity level if the severity level of the first controller is relatively higher than the defined maximum severity level (As seen in the rejection of Claim 2 above, the child does not have to inherit the value of the parent, so it is conceivable that the parent has logging set to true and the child has logging set to false, which would mean the parent has a higher logging lever than the child, and the child will use its lower level rather than inherit the parent value)".

31. With respect to Claim 22, the combination of Shah, Ullman and Biegel disclosed: "wherein said first and second controllers are tracing controllers (Shah, [0066], lines 8-10, where the child trace node can be configured independently of its parent trace node), further comprising: receiving a trace message directed at the second controller (Shah, [0042], lines 2-4, and Figure 6, object 601, being the trace logger); comparing a severity value associated with the trace message with the severity level of the second controller (Shah, [0050], lines 1-2 and 6-8); and processing the trace message if the severity value associated with the trace message is greater than or equal to the severity level of the second controller (Shah, [0050], lines 2-4).

32. With respect to Claim 23, the combination of Shah, Ullman and Biegel disclosed: "wherein processing comprises formatting information contained in the trace message (Shah, [0051], lines 1-2) and forwarding the formatted information to a specified output destination (Shah, [0043], lines 1-5)".

33. With respect to Claim 24, the combination of Shah, Ullman and Biegel disclosed: "wherein the trace message is emitted in response to the execution of a specified area of program code within an application (Shah, [0042], lines 6 and second Col., lines 1-6)".

34. With respect to Claim 25, the combination of Shah, Ullman and Biegel disclosed: "wherein said first and second controllers are logging controllers (Shah, [0066], lines 8-10, where the child trace node can be configured independently of its parent trace node, and [0068], lines 1-3, where message loggers can be similarly configured) further comprising: receiving a log message directed at the second controller (Shah, [0042], lines 2-3); comparing a severity value associated with the log message with the severity level of the second controller (Shah, [0050], lines 1-2, and 6-8); and processing the log message if the severity value associated with the trace message is greater than or equal to the severity level of the second controller (Shah, [0050], lines 2-4)".

35. With respect to Claim 26, the combination of Shah, Ullman and Biegel disclosed: "wherein said first and second controllers are tracing controllers (Shah, [0066], lines 8-10, where the child trace node can be configured independently of its parent trace node) processing comprises formatting information contained in the log message (Shah, [0051], lines 1-2) and forwarding the formatted information to a specified output destination (Shah, [0043], lines 1-5)".

36. With respect to Claim 27, the combination of Shah, Ullman and Biegel disclosed: "wherein the second controller is associated with a particular category (Shah, [0042], lines 4-5) and wherein the log message is from an application and/or network component associated with the category (Shah, [0042], lines 4-5)".

Response to Arguments

37. Applicant's arguments with respect to claims 1-30 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MATTHEW S. LINDSEY whose telephone number is (571)270-3811. The examiner can normally be reached on Mon-Thurs 7-5, Fridays 7-12.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Follansbee can be reached on (571) 272-3964. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MSL
12/18/2008

/John Follansbee/

Supervisory Patent Examiner, Art Unit 2451